



Descripteurs locaux de forme pour la reconnaissance du scripteur

I. Siddiqi, N. Vincent

► To cite this version:

I. Siddiqi, N. Vincent. Descripteurs locaux de forme pour la reconnaissance du scripteur. Colloque International Francophone sur l'Écrit et le Document, Oct 2008, France. pp.157-162. hal-00334412

HAL Id: hal-00334412

<https://hal.science/hal-00334412>

Submitted on 26 Oct 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Descripteurs Locaux de Forme pour la Reconnaissance de Scripteur

Imran Siddiqi – Nicole Vincent

Laboratoire CRIP5-SIP
Université Paris Descartes, 75006 Paris, France

{siddiqi;nicole.vincent}@math-info.univ-paris5.fr

Résumé : Cette communication présente une méthode efficace pour l'identification et la vérification de scripteur basée sur les documents manuscrits. Nous avons développé une approche locale reposant sur l'idée que dans un texte manuscrit, il existe des formes redondantes qu'un scripteur utilise fréquemment comme il l'écrit. Ces formes pourraient être exploitées pour identifier l'auteur d'un document ou vérifier la paternité d'un document. Pour extraire ces formes, l'écriture est divisée en un grand nombre d'imagettes, un ensemble de descripteurs est extrait de chacune et ensuite les imagettes similaires sont groupées dans des classes. Les formes qui se produisent fréquemment pour un auteur sont ainsi extraites. L'auteur du document inconnu est identifié en utilisant une décision Bayésienne. Le système, évalué sur une base de 55 scripteurs, conduit à des résultats encourageants. .

Mots-clés : Analyse de l'écriture manuscrite, Identification, Vérification, Classification, Décision Bayésienne.

1 Introduction

Malgré les prédictions d'un monde sans papier et le développement des documents électroniques, les documents manuscrits ont gardé leur importance et les problèmes de l'identification et de l'authentification des auteurs ont constitué un domaine de recherche actif au cours de ces dernières années. Comparé au texte électronique ou imprimé, le texte manuscrit diffuse des informations additionnelles sur la personnalité de la personne qui a écrit. Il existe un certain degré de stabilité dans le modèle d'écriture d'un individu, ce qui permet d'identifier l'auteur pour lequel on a déjà vu un texte écrit.

La nécessité d'authentifier un texte écrit est un problème récurrent, non seulement du point de vue de la biométrie comportementale [BEN05, PLA89, SER02, SRI02], mais également dans le contexte de la reconnaissance d'écriture [NOS99] en exploitant le principe de l'adaptation du système au type du scripteur. La recherche dans le domaine a beaucoup évolué dans les années récentes et une grande variété de techniques ont été proposées qui peuvent être classées soit comme globales [BOU98, SAI00] : basées sur l'aspect global de l'écriture, soit locale [BEN05, SHM04, SID07], qui identifient l'auteur à partir de l'extraction des caractéristiques locales de l'écriture, qui sont inhérentes à la manière dont un scripteur écrit spécifiquement des

caractères. La combinaison de caractéristiques à différents niveaux est également connue pour améliorer le taux d'identification [BUL07]. L'identification de scripteur basée sur la reconnaissance d'écriture a également été proposée dans la littérature [SCH04, SCH06].

La présente communication est basée sur une approche locale, présentant un système de reconnaissance de scripteur. Notre méthode repose sur la redondance des formes qui est aussi la base de la méthode proposée par [BEN05]. En effet, leur approche est liée à la façon dont les lettres sont tracées et segmentées comme si le but était de lire le texte. Nous pensons que la reconnaissance du scripteur est indépendante de ce qui est écrit et est plutôt liée à la manière physique par laquelle des lignes ou des boucles sont produites. Ainsi l'échelle de l'observation peut être inférieure à celle d'une lettre. Aucune interprétation sémantique des portions de trait analysé n'est nécessaire.

Nous présentons dans les paragraphes qui suivent les différentes étapes nécessaires à la reconnaissance du scripteur. De manière classique, notre système peut être décomposé en deux phases : l'apprentissage de chaque scripteur autorisé, en extrayant les caractéristiques de chacun, il s'effectue lors de l'enrôlement du scripteur, et la reconnaissance de scripteur appliquée sur un document inconnu.

2 Apprentissage des Scripteurs

Depuis quelques années, la tendance des recherches sur les écritures manuscrites, s'est focalisée vers l'extraction de caractéristiques propres au scripteur. La méthode que nous proposons s'inscrit dans ce cadre. Il ne s'agit pas de calculer la valeur d'une caractéristique pour un scripteur mais d'extraire de l'écriture ses caractéristiques propres. Nous développons une approche locale cherchant de mettre en évidence des détails fréquents dans une écriture. Nous montrons qu'il existe, dans un texte manuscrit, des formes redondantes pouvant être exploitées par un système d'identification du scripteur. Ces formes ont été extraites par un découpage adaptatif de l'écriture en imagettes, l'extraction de descripteurs de chacune des imagettes puis une classification groupant les imagettes similaires. Ces étapes sont présentées en détail dans les sections suivantes.

2.1 Découpage de l'écriture

Le découpage de l'écriture est utilisé pour extraire des éléments inhérents au scripteur, donc c'est une partie importante du processus. Il doit être dépendant du tracé pour que les contenus puissent être comparables. On a choisi un découpage en carrés de taille $n \times n$. La technique la plus simple consisterait à découper la totalité de l'image régulièrement de gauche à droite et de bas en haut. Mais cette technique conduit à un grand nombre d'images (contenant des pixels textes) mais les carrés ne sont pas toujours bien positionnés par rapport aux traits, certains traits sont divisés dans leur largeur entre plusieurs images différentes. Dans nos travaux précédents [SID07], nous avons présenté une approche plus adaptative en adaptant le découpage composante connexe par composante connexe. Pour chaque composante connexe dans le texte, nous fixons l'origine verticale et déplaçons chaque fenêtre de gauche à droite pour trouver sa meilleure position.

Ici, nous présentons une nouvelle technique, en suivant la direction naturelle de la main et du tracé. Il ne s'agit pas de retrouver l'ordre exact du tracé mais de positionner les images par rapport aux éléments caractéristiques fondamentaux. De manière à trouver les extrémités du trait, nous nous référons au squelette des différentes composantes connexes, et positionnons une première fenêtre sur un point extrême du squelette.

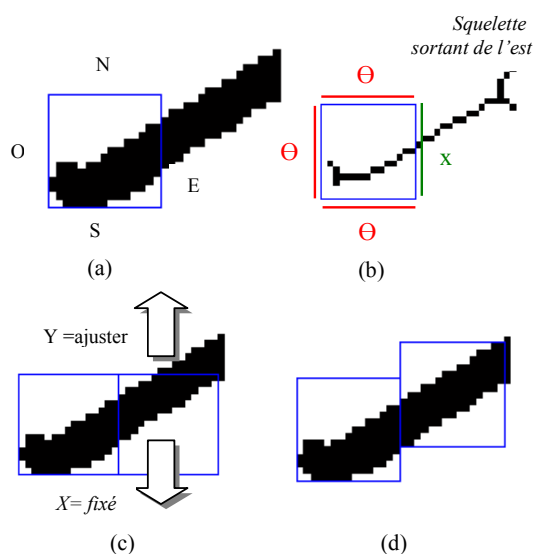


FIG. 1 – Placement de fenêtre (a) début d'un trait, (b) choix d'un côté avec trait sortant, (c) positionnement initial de la fenêtre suivante, (d) glissement de la fenêtre par rapport au trait.

Pour chaque fenêtre nous définissons quatre drapeaux : *Est*, *Ouest*, *Nord* et *Sud* ; Le drapeau étant *affiché* si le trait sort de l'image par ce côté. Si le squelette sort de E (ou de O), nous plaçons la prochaine fenêtre vers la droite (gauche respectivement) de la position présente (sur l'image originale), et déplaçons la fenêtre dans la direction verticale (en haut ou en bas) pour trouver le meilleur placement suivant. Par contre, s'il sort de N (ou de S), nous plaçons la prochaine fenêtre

au-dessus (en dessous respectivement), et déplaçons la fenêtre dans la direction horizontale (gauche ou droite) de sorte qu'elle soit bien placée. Le processus est illustré dans la figure 1.

Dans le cas où le squelette sortirait de plus d'une direction, nous traitons séparément chacune des directions concernées. L'algorithme du placement des fenêtres est appliqué à chacune des composantes et par conséquent nous avons un découpage de l'écriture en petites fenêtres; illustré pour le mot 'headlines' dans la figure 2 .

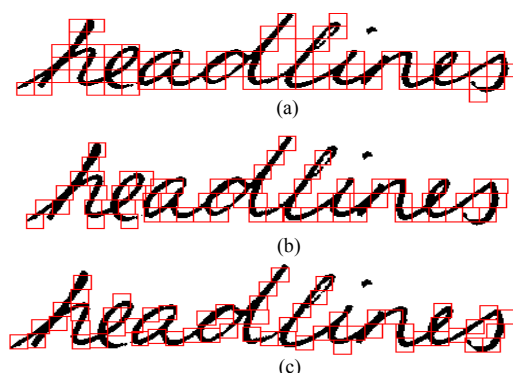


FIG. 2 – Découpage du mot 'headlines' par (a) Division Régulière (b) Déplacement horizontal de fenêtres (c) Positionnement de fenêtres relatif au trait

La taille des fenêtres utilisées pour le découpage de l'écriture est un paramètre important de notre méthode. La taille est choisie en fonction de deux critères. Le premier est que la taille doit être assez grande pour contenir assez d'informations sur le style du scripteur et le second est que cette même taille doit être assez petite pour assurer une bonne performance du critère d'identification. Pour nos évaluations, nous avons testé les tailles de 11x11, 13x13, 15x15 et 17x17. Parmi ces dernières, une taille de 13x13 a donné les meilleurs résultats [SID07] ; ainsi pour le reste de la discussion, la taille de la fenêtre est fixée à 13.

2.2 Extraction des Caractéristiques

Une fois que le texte est divisé en images, nous procédons à l'extraction de descripteurs de forme sur chacune. La position de la trace dans la fenêtre est d'abord normalisée afin que les caractéristiques calculées soient invariantes à la *translation*. Il n'est cependant pas souhaitable dans notre cas d'avoir: i) l'invariance à l'échelle: alors que nous ne prévoyons pas que le scripteur modifie la taille de l'écriture dans un même échantillon d'écriture et ii) l'invariance à la *rotation*: comme une forme et sa version tournée ne sont pas produites par le même geste de la main.

L'ensemble des descripteurs que l'on calcule pour chaque image comprend:

- La projection horizontale: Le nombre de pixels de texte dans chaque ligne de l'image.

- La Projection Verticale: Le nombre de pixels de texte dans chaque colonne de l'imagette.
- Profil Supérieur: La distance, sur chaque colonne, entre le haut de chaque imagette et le premier pixel du texte.
- Profil Inférieur: La distance, sur chaque colonne, entre le haut de chaque imagette et le dernier pixel du texte.
- Orientation: L'orientation générale de la forme: la mesure de l'angle comprise entre -90° et 90° entre l'axe des abscisses et le grand axe de l'ellipse d'inertie de la forme.
- Excentricité: Le ratio entre la longueur de la plus longue corde de la forme et la plus longue perpendiculaire à cette corde.
- Rectangularité: Le ratio aire de l'objet / aire de la bounding box.
- Allongement: Le ratio entre la hauteur et la largeur du rectangle minimal circonscrit à la forme (les côtés ne sont pas nécessairement horizontaux et verticaux).
- Périmètre: Le nombre de pixels dans le contour de la forme.
- Irrégularité : la proportion des pixels de l'enveloppe convexe, qui sont aussi dans la région, elle est calculée comme : Aire / Aire Convexe.

La figure 3 illustre les projections horizontale et verticale et les profils calculés pour une imagette.

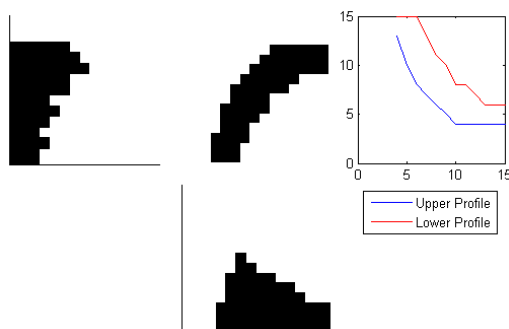


FIG. 3 – Les projections et les profils calculés sur une imagette

Une fois que tous les descripteurs ont été calculés, les valeurs sont normalisées dans l'intervalle $[0,1]$ et, par conséquent, chaque imagette est représentée par un vecteur de dimension $d = 4n + 6$, où n est la taille de la fenêtre. Dans notre cas, soit S l'ensemble des imagettes, nous avons donc:

$$S = \{S_i\}, \text{ avec chaque } S_i = (s_i^1, \dots, s_i^d)$$

2.3 Regroupement des imagettes

Représentant chaque imagette par un vecteur, on procède ensuite à leur regroupement de manière à réduire la quantité de données et à rendre le résultat indépendant de la quantité de texte étudié. L'objectif est de grouper les formes produites par le même geste de la main dans les mêmes classes. Pour notre classification, nous avons évalué un certain nombre d'algorithmes qui ne demandent pas un choix préalable du nombre de

classes. La dissimilarité entre deux formes est calculée par la distance euclidienne définie dans l'espace des caractéristiques :

$$d(S_i, S_j) = \sqrt{\sum_{k=1}^d (s_{i,k} - s_{j,k})^2}$$

Nous allons maintenant présenter brièvement les algorithmes de regroupement employées :

2.3.1 Classification Séquentielle

Nous choisissons un seuil de proximité et commençons par la première imagette comme le centre de la première classe. Pour chacune des formes suivantes, nous calculons la distance euclidienne entre l'élément courant et le centre de chaque classe. L'élément est alors attribué au groupement le plus proche. Au cas où il ne se rapproche pas assez d'un groupement, un nouveau groupe est créé [SID07]. Le problème, toutefois, c'est que la procédure est sensible à l'ordre dans lequel les modèles sont présentés. Nous avons donc modifié l'approche.

2.3.2 Classification Séquentielle Multi-Phase

Pour résoudre ce problème de la classification séquentielle, [BEN 05] propose d'effectuer plusieurs phases de classification avec une sélection aléatoire des imagettes afin d'être moins sensible aux conditions initiales. Chacune des phases de regroupement prévoit donc un nombre variable de classes. Les dernières classes sont définies comme les groupes d'imagettes qui sont toujours regroupées au cours de chaque phase. Les autres (imagettes qui ne font pas partie d'un groupe) sont ensuite affectés à la classe la plus proche:

$$Class(S_j) = \arg \min_i \left(\sqrt{S_j - \mu_i}^T \sum_i^{-1} (S_j - \mu_i) \right)$$

$$\forall S_j \in L$$

Où L représente l'ensemble des imagettes non classées, μ_i est la moyenne de cluster i et Σ_i sa matrice de covariance.

2.3.3 Classification Séquentielle à deux étapes

Nous proposons maintenant un algorithme de classification en deux étapes qui pourrait être considéré comme hybride d'une classification par partition et d'une classification séquentielle. Nous divisons l'ensemble S en Φ sous-ensembles disjoints (Sp_1, \dots, Sp_Φ) , en fixant un critère portant sur la dimension k du vecteur de caractéristiques. Chacune de ces Φ partitions est ensuite regroupée séparément en utilisant l'algorithme séquentiel défini ci-dessus et les résultats de ces procédures de classification sont fusionnés pour obtenir l'ensemble final de clusters. Dans notre cas, nous avons choisi de diviser les imagettes en fonction de l'orientation du trait, en divisant l'intervalle -90° à 90° en quatre parties ($\Phi=4$).

2.3.4 Classification par arbre de recouvrement minimal

Nous présentons maintenant le graphe sur lequel sera recherché le recouvrement minimal. Un arbre de

recouvrement minimal (*Minimum Spanning Tree* MST) d'un graphe pondéré relie tous les points de données au moindre coût possible. Du point de vue de la classification: si les poids des arêtes représentent les distances entre les points de données, la suppression des arêtes du MST conduit à une collection de composantes connexes qui peuvent être considérées comme les classes [PAI05]. Pour notre ensemble de sous-images, nous définissons le graphe (pondéré, non orienté) $G(S) = (V, E)$, comme suit: l'ensemble des sommets $V = \{S_i \mid S_i \in S\}$ et celui des arêtes $E = \{(S_i, S_j) \mid \text{pour } S_i, S_j \in S \text{ et } i \neq j\}$. Chaque arête $(u, v) \in E$ a un poids qui représente la distance (euclidienne) $d(u, v)$, entre u et v . Un arbre de recouvrement T du graphe $G(S)$ est un sous-graphe connecté de $G(S)$ tel que T contienne tout sommet de $G(S)$, et celui avec la distance totale minimum est le MST. La procédure de regroupement peut être résumée par :

- Construire un graphe connecté G à partir de S
- Construire un arbre de recouvrement minimum T de G
- Retirez toutes les arêtes avec des poids supérieurs à un seuil
- conserver les composantes connexes comme les classes.

Le MST est construit en utilisant l'algorithme de Prim et la classification est basée sur l'idée que deux points de données proches devraient faire partie de la même classe (sous-arbre) et des points de données éloignés devraient appartenir à des classes différentes et donc l'arête doit être coupée. Le nombre de classes obtenues, naturellement, est sensible à la valeur de seuil choisie. L'algorithme fonctionne plutôt bien à condition que les distances inter-classes sont nettement plus grandes que les distances intra-classe.

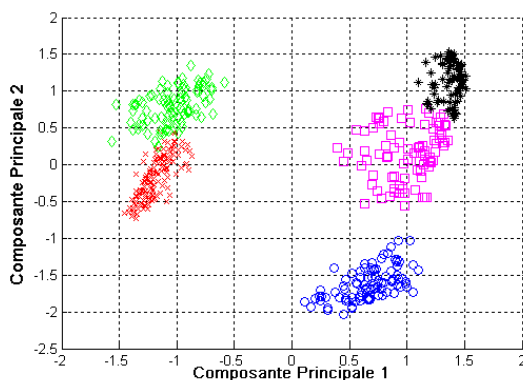


FIG. 4 – Projection des 5 classes les plus fréquentes dans le plan formé par les deux premières composantes principales

Vu l'efficacité des caractéristiques proposées, Fig. 4 montre les clusters obtenus après avoir effectué une Analyse en Composantes Principales (ACP), réduisant la dimensionnalité à 2, ce qui montre que les classes sont assez séparables en utilisant ces caractéristiques.

Une fois les imagerie regroupées (par l'une des méthodes examinées ci-dessus), nous pouvons trier les classes selon leur cardinal et ne conserver que celles qui ont suffisamment d'éléments. Le terme suffisant, toutefois, est relatif et nous en trouverons la valeur de manière automatique en gardant les top M plus importantes classes qui permettent de couvrir 90% des pixels du texte dans l'image. Nous voyons déjà dans la figure 5 que le nombre de classes retenues (M) pour deux échantillons du même scripteur est assez proche, il pourrait servir de paramètre utile dans la phase d'identification, comme nous le verrons plus loin.

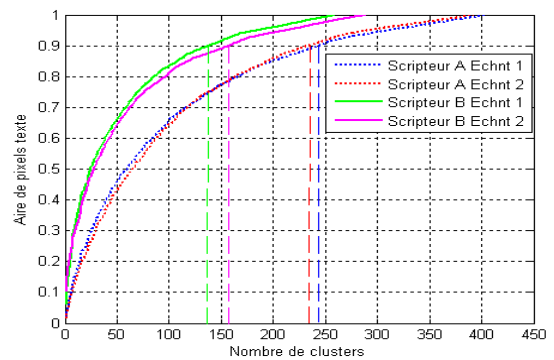


FIG. 5 – Nombre de classes en fonction de l'aire de texte couvert en pixels

Suite à ce filtrage des classes, le document D est représenté par un ensemble de classes C .

$$C^D = \{C_k \mid k \leq M\}$$

Pour chaque classe C_k , nous avons:

$$C_k = \{S_{1,k}, \dots, S_{m,k}\}, m = \text{card}(C_k)$$

$$\text{Et chaque } S_{i,k} = (s_{i,k}^1, \dots, s_{i,k}^d)$$

On calcule aussi pour chaque classe, la probabilité de son occurrence $P(C_k)$, sa matrice de covariance Cov_k et la valeur moyenne du vecteur \bar{S}_k , représentant ainsi le document D comme:

$$D^r = \{F_k, k \leq \text{card}(C)\}$$

$$\text{où } F_k = \{P(C_k), Cov_k, \bar{S}_k\}$$

3 Reconnaissance du Scripteur

La reconnaissance du scripteur comprend l'identification ainsi que la vérification. Le problème de l'identification consiste à faire correspondre une écriture donnée à une écriture de référence dont on connaît l'auteur. Cette association est réalisée en fonction d'un indice de similarité. D'autre part, la tâche de vérification du scripteur consiste à authentifier le scripteur d'un document.

3.1 Identification

Pour pouvoir identifier l'auteur d'un document inconnu, on commence par le découpage de l'écriture, suivi par la classification des imagerie. Ces étapes sont

identiques à celles effectuées pendant la phase d'apprentissage. L'image à tester T , est alors représentée par une ensemble des vecteurs moyennes de chaque classe: $T^r = \{\bar{S}_j\}$. Si la différence entre le nombre de classes de T et le document de référence D est au-dessus d'un certain seuil (fixé à 0.35), nous rejetons ce document et nous passerons au prochain document de la base de référence:

$$\frac{|card(C^T) - card(C^D)|}{card(C^T)} < \eta$$

Si un document de référence répond à cette condition, nous passerons à l'étape suivante définissant la mesure de similarité entre le document à tester T représenté par T^r et un document de référence D représenté par D^r par :

$$SIM(T, D) = \frac{1}{card(T^r)} \sum_{j=1}^{card(T^r)} \max_{C_i \in C^D} (P(C_i | \bar{S}_j))$$

C'est à dire: pour chaque \bar{S}_j dans T^r , l'objectif est de trouver la classe i du document D qui maximise la probabilité de \bar{S}_j appartenant à la classe C_i .

$P(C_i | \bar{S}_j)$ est estimée par:

$$-\frac{1}{2} \log |Cov_i| - \frac{1}{2} (\bar{S}_j - \bar{S}_i)^T Cov_i^{-1} (\bar{S}_j - \bar{S}_i) + \log P(C_i)$$

où

\bar{S}_i = la moyenne de classe i (dx1 vecteur)

Cov_i = la matrice de covariation de classe i (dx dx matrice)

Avec l'hypothèse que les éléments d'une classe ont une distribution normale, une dérivation de l'expression ci-dessus peut être trouvée dans [SID07]. Nous calculons l'indice de similarité entre le document T et tous les documents dans la base de référence R et identifions l'auteur du document en question comme étant l'auteur du document, qui maximise l'index.

3.2 Vérification

La vérification du scripteur est réalisée dans le cadre classique du test de Neyman-Pearson. En faisant varier le seuil de décision sur l'indice de similarité calculé entre les deux documents, les courbes «Receiver Operating Characteristic» (ROC) sont calculées. La performance de la vérification est quantifiée par la «Equal Error Rate» (EER), qui est le point où la False Acceptation Rate est égale à la False Rejection Rate.

4 Résultats

Pour l'apprentissage et l'évaluation de notre système nous avons aléatoirement choisi 55 scripteurs de la base IAM [MAR99] qui contient des textes manuscrits sans contrainte. Une image de l'écriture de chacun des

scripteurs est utilisée dans la phase d'apprentissage et une autre est utilisée dans les tests. Chaque image contient en moyenne, 8-10 lignes de texte. Nous avons d'abord effectué une série exhaustive d'expériences pour trouver la meilleure valeur de seuil pour chacune des méthodes de classification présentée. Les taux obtenus en identification sont ensuite comparés comme illustré dans la figure 6.

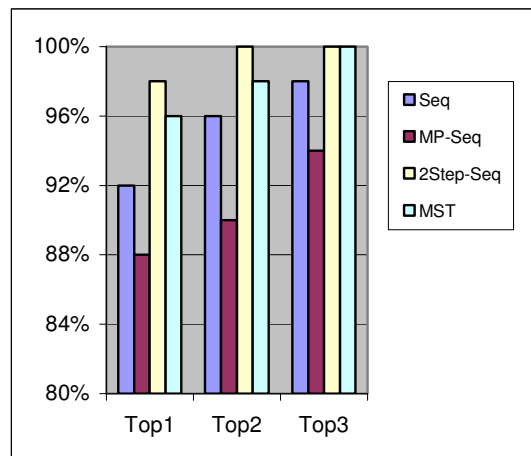


FIG. 6 – Comparaison de résultats d'identification en fonction de la méthode de classification utilisée

Comme on peut le constater à partir de la comparaison des résultats, la classification séquentielle à deux étapes surpasse les autres méthodes, réalisant un taux d'identification de 98%. La classification MST se comporte presque aussi bien (96%), mais le calcul des MST est assez inefficace. En outre, dans les cas où les différences entre les distances inter-classes et intra-classe ne sont pas très importantes, les résultats sont susceptibles de se dégrader. La classification séquentielle multiphase, même si elle produit des classes très fines, se situe derrière les autres méthodes en terme de performance.

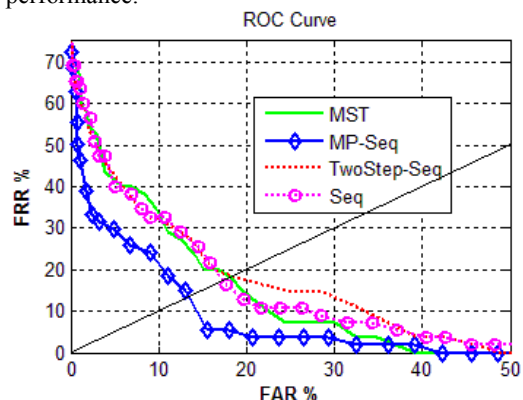


FIG. 7 – Comparaison de résultats sur la vérification

Pour la tâche de vérification du scripteur, on calcule les courbes ROC pour les différentes méthodes de classification. Contrairement à ce que l'on a pu observer dans le cas du problème d'identification, c'est la classification séquentielle multiphase qui obtient le meilleur résultat (un EER de l'ordre de 13%).

Clairement, l'efficacité du système sur la vérification du scripteur n'est pas aussi bonne que sur l'identification et doit être améliorée. Une possibilité pourrait être d'avoir un seuil de vérification qui dépend du scripteur, au lieu d'utiliser une valeur globale de seuil pour l'ensemble des scripteurs de la base de référence. Une autre option pourrait être d'introduire certains post-traitements quand la différence entre l'indice de similarité du meilleur match et celui du second est inférieur à un certain seuil. Ces problèmes seront abordés dans la recherche à suivre. Le système doit également être testé sur une plus grande base, permettant une comparaison avec les autres méthodes qui ont été évaluées sur le même ensemble de données.

5 Conclusion

Nous avons présenté une méthode efficace pour la reconnaissance de scripteurs à partir de leur écriture manuscrite. La technique s'appuie sur l'extraction des formes qu'un auteur spécifique emploie fréquemment quand il dessine les caractères. Cependant, le texte présenté doit être suffisamment long pour qu'il nous permette d'exploiter l'idée de la redondance de formes dans une écriture. Les taux d'identification réalisés sont très encourageants et ils permettent de valider notre méthode. Les résultats de la vérification, toutefois, ne sont pas aussi bons et doivent être améliorés, ce qui fera l'objet de nos futures recherches. De plus, en changeant la taille de fenêtre pendant la phase de découpage de l'écriture, cette méthode peut s'appliquer à des textes autres que des textes occidentaux, tels que ceux en écriture asiatique ou arabe. On peut avoir un système plus robuste, le rendant capable d'ajuster automatiquement la taille de la fenêtre en fonction des détails de l'écriture. La capacité d'analyser le texte à différentes résolutions et à différentes épaisseurs des traits peut également être améliorée.

Références

- [BEN 05] A. Bensefia, T. Paquet and L. Heutte: A writer identification and verification system, *Pattern Recognition Letters*, Elsevier Science Inc. New York, USA, Vol 26, issue 13, 2005, pp. 2080-2092.
- [BOU 98] V. Bouletreau, N. Vincent, R. Sabourin, H. Emptoz: Handwriting and signature: one or two personality identifiers?, *In Proc. of Fourteenth International Conference on Pattern Recognition*, Brisbane, vol.2, 1998, pp. 1758-1760.
- [BUL 07] M. Bulacu and L. Schomaker: Text-Independent Writer Identification and Verification Using Textural and Allographic Features, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007, pp. 701-717.
- [MAR 99] U. Marti and H. Bunke: A full English sentence database for off-line handwriting recognition, *In Proc. of 5th International Conference on Document Analysis and Recognition*, Bangalore, India, 1999, pp. 705-708.
- [NOS 99] A. Nosary, L. Heutte, T. Paquet, Y. Lecourtier: Defining writer's invariants to adapt the recognition task, *In Proc. of 5th International Conference on Document Analysis and Recognition*, Bangalore, India, 1999, pp. 765-768.
- [PAI 05] N. Päävinen: Clustering with a minimum spanning tree of scale-free-like structure, *Pattern Recognition Letters*, Vol. 26, Issue 7, Elsevier Science Inc. New York, USA, 2005, pp. 921-930.
- [PLA 89] R. Plamondon and G. Lorette: Automatic signature verification and writer identification – the state of the art, *Pattern Recognition*, vol. 22, n°2, 1989, pp. 107-131.
- [SAI 00] H.E.S. Said, T.N Tan, K.D. Baker: Personal Identification Based on Handwriting, *Pattern Recognition*, vol. 33, 2000, pp. 149-160.
- [SCH 04] A. Schlapbach and H. Bunke: Off-line handwriting identification using HMM based recognizers, *In Proc. of the 17th International Conference on Pattern Recognition*, Cambridge, United Kingdom, August 2004, pp. 654-658.
- [SCH 06] A. Schlapbach and H. Bunke: Off-line Writer Identification Using Gaussian Mixture Models, *In Proc. of the 18th International Conference on Pattern Recognition*, Hong Kong, August 2006, pp. 992-995.
- [SER 02] A. Seropian and N. Vincent: Writers Authentication and Fractal Compression, *In Proc. Of the Eighth International Workshop on Frontiers in Handwriting Recognition*, 2002.
- [SHM 04] L. Schomaker and M. Bulacu: Automatic Writer Identification Using Connected-Component Contours and Edge-Based Features of Uppercase Western Script, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004, pp. 787-798.
- [SID 07] I. Siddiqi and N. Vincent: Writer Identification in Handwritten Documents, *In Proc. of the 9th Int'l conference on Document Analysis and Recognition (ICDAR 07)*, Curitiba, Brazil, 2007, pp. 108-112.
- [SRI 02] S. Srihari, S. Cha, H. Arora, and S. Lee: Individuality of handwriting, *J. of Forensic Sciences*, 47(4), 2002, pp. 1-17.
- [MAR 99] U. Marti and H. Bunke: A full English sentence database for off-line handwriting